Classification

Ruwen Qin
Stony Brook University

November 10, 2025

Data used in this module:

- ai4i2020.csv

Python notebook used in this module

- Classification.ipynb

1 Introduction

Considering an entity characterized by an N dimensional feature representation: $\boldsymbol{X} = [X_1, \ldots, X_N]^{\mathrm{T}}$. We would like to infer a target variable of interest, Y, which takes values from a finite set of K categories, $\{c_1, \ldots, c_K\}$. Classification is about developing a classifier: h: $\boldsymbol{X} \rightarrow Y$, which can assign an observation with feature values $\boldsymbol{x} = [x_1, \ldots, x_N]^{\mathrm{T}}$ to a class label y.

There are many classification problems in the real world. For example, given an image captured by a robot in infrastructure inspection, a classifier can determine if the image contains defect. Given measures of machine attributes in operations, a classifier can predict the type of failure the machine will confront. This learning module is developed based on related chapters in [1, 2, 3, 4, 5].

1.1 Bayes Classifier

Define the probability that the class of the response is c_k conditional on the observed feature values is:

$$p_k = P(c_k|\boldsymbol{x}), \tag{1}$$

and $\sum_{k=1}^{K} p_k = 1$.

According to Bayes' theorem,

$$p_k = \frac{f_k(\boldsymbol{x})\pi_k}{\sum_{r=1}^{K} f_r(\boldsymbol{x})\pi_r} \tag{2}$$

where π_k is the prior probability:

$$\pi_k = P(c_k), \tag{3}$$

and $f_k(\boldsymbol{x})$ is the likelihood probability,

$$f_k(\boldsymbol{x}) = P(\boldsymbol{x}|c_k). \tag{4}$$

The Bayes classification rule says that the optimal rule is one that picks the class with the maximum posterior probability:

$$h^*(\boldsymbol{x}) = \arg\max_k p_k = \arg\max_k f_k(\boldsymbol{x})\pi_k. \tag{5}$$

1.2 Three Approaches to Classification

The Bayes rule depends on unknown quantities $f_k(\boldsymbol{x})$ and π_k for $k = 1, \ldots, K$. Therefore, we need to use the data to find some approximation to the Bayes rule.

Generally speaking. there are three major approaches:

- The first approach is density estimation. We use data to estimate the prior and likelihood probabilities and then compute the posterior probability using the Bayes's theorem.
- The second approach is to model the conditional probability p_k directly, for example, as a parametric model.
- The last approach involves constructing a discriminant function that directly assign each vector to a specific class.

Classifiers can be developed in one of these approaches or a mix of them. In this learning module, we will introduce several classification methods.

2 Classification Performance Metrics

2.1 Binary Classification Results

Assume that we have a test dataset to evaluate the performance of a classification model. There are four scenarios of classification result:

- *True Positive (TP)*: A true positive instance is correctly predicted as positive;

- *True Negative (TN)*: A true negative instance is correctly predicted as negative;
- *False Positive (FP)*: A true negative instance is mistakenly predicted as positive;
- *False Negative (FN)*: A true positive instance is mistakenly predicted as negative;

2.2 Confusion Matrix

The confusion matrix in Table 1 counts the numbers of instances according to their true label and predicted label.

Table 1: Confusion Matrix

		predicted label ($\widehat{y}$)		
		1	...	K
true label (y)	1			
	$\vdots$			
	K			

2.3 Error Rates

Overall Error Rate is the percentage of misclassified observations:

$$\text{Overall Error Rate} = \frac{\sum_{i=1}^{M} 1\{y_i \neq \widehat{y}_i\}}{M}. \tag{6}$$

Overall error rate conveys an aggregate measure of misclassification, it counts a false positive the same as a false negative. To account for the asymmetric costs in misclassification, we define error rate with respect to the individual classes.

Class k error rate is the percent of class k instances that are misclassified:

$$\text{error rate}_k = \frac{\sum_{i=1}^{M} 1\{y_i = k\}1\{\widehat{y}_i \neq k\}}{\sum_{i=1}^{M} 1\{y_i = k\}}. \tag{7}$$

2.4 Correct Rates

Accuracy is the percentage of correctly classified observations:

$$\text{Accuracy} = \frac{\sum_{i=1}^{M} 1\{y_i = \widehat{y}_i\}}{\sum_{i=1}^{M} 1\{y_i\}}. \tag{8}$$

Accuracy is 1-overall error rate.

Recall is the ability to correctly predict class k instances, and it is calculated as:

$$\text{Recall}_k = \frac{\sum_{i=1}^{M} 1\{\widehat{y}_i = y_i = k\}}{\sum_{i=1}^{M} 1\{y_i = k\}}. \tag{9}$$

Precision is the proportion of instances predicted to be class k that are actually in class k:

$$\text{Precision}_k = \frac{\sum_{i=1}^{M} 1\{\widehat{y}_i = y_i = k\}}{\sum_{i=1}^{M} 1\{\widehat{y}_i = k\}}. \tag{10}$$

F1 Score combines precision and recall into a single measure and is defined as:

$$\text{F1}_k = \frac{2}{\frac{1}{\text{Recall}_k} + \frac{1}{\text{Precision}_k}}. \tag{11}$$

2.5 Precision-Recall Curve and Average Precision (AP)

Precision and recall for each class are dependent of the threshold for translating the probabilistic prediction into class prediction. By changing the threshold value, we can get different pairs of precision and recall. Accordingly, a *precision-recall curve* can be created for displaying the trade-off between them, where x-axis is the recall and the y-axis is the precision. The *Average Precision* (AP_k) is the area under that curve.

For binary classification, the precision-recall curve is created for the positive class (i.e., class 1). For multiclass classification, this curve is created for every class.

2.6 Receiver Operating Characteristic (ROC) and Area Under ROC (AUC)

Receiver Operating Characteristic (ROC) curve is a graphical approach for displaying the tradeoff between a classifier's ability to correctly identify class k instances and the error rate for misclassifying instances of other classes as class k instances. The x-axis is the false positive rate for class k and the y-axis is Recall_k. The curve is obtained by changing the threshold value for classification. The area under the ROC curve is called AUC_k and measures the quality of the classifier in classifying class k. The larger the AUC_k, the better the classifier performs.

For binary classification, the ROC curve is created for the positive class (i.e., class 1), with x-axis being the class 0 error (i.e., false positive rate) and y-axis being the recall for the positive class (i.e., the true positive rate). For multiclass classification, this curve is created for every class.

2.7 Macro-level Metrics

A macro-level average is an unweighted average of class-level metric values:

$$\mathrm{mRecall} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Recall}_k \tag{12}$$

$$\mathrm{mPrecision} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Precision}_k \tag{13}$$

$$\mathrm{mF1} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{F1}_k \tag{14}$$

$$\mathrm{mAP} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{AP}_k \tag{15}$$

$$\mathrm{mAUC} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{AUC}_k \tag{16}$$

2.8 Weighed Metrics

A weighted average is about calculating the class-level metric values and then averaging them with the weights determined according to class support (i.e., the number of true instances for each class):

$$\mathrm{mRecall} = \sum_{k=1}^{K} \frac{\sum_{i=1}^{M} 1\{y_i = k\}}{M} \mathrm{Recall}_k \tag{17}$$

$$\mathrm{mPrecision} = \sum_{k=1}^{K} \frac{\sum_{i=1}^{M} 1\{y_i = k\}}{M} \mathrm{Precision}_k \tag{18}$$

$$\mathrm{mF1} = \sum_{k=1}^{K} \frac{\sum_{i=1}^{M} 1\{y_i = k\}}{M} \mathrm{F1}_k \tag{19}$$

$$\text{mAP} = \sum_{k=1}^{K} \frac{\sum_{i=1}^{M} 1\{y_i = k\}}{M} \text{AP}_k \tag{20}$$

$$\text{mAUC} = \sum_{k=1}^{K} \frac{\sum_{i=1}^{M} 1\{y_i = k\}}{M} \text{AUC}_k \tag{21}$$

2.9 Micro Average

To find a micro average, we calculate metrics globally. The accuracy in (8) and the overall error rate in (6) are both micro averages.

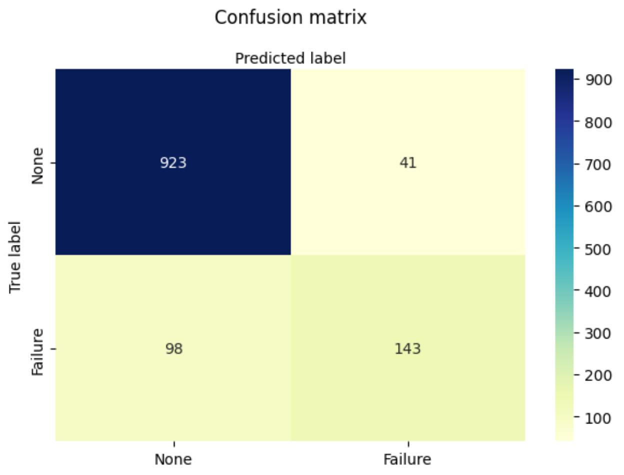
3 Examples

3.1 An Example of Binary Classification

Let's take a look at the binary classification problem in the Python Notebook. Results below is obtained using an LDA classifier.

Table 2 is the confusion matrix given the cutoff point is 0.5.

Table 2: Confusion matrix for the concrete defect class prediction



Below is the report of performance metrics:

```

 Class-level report:
              precision    recall  f1-score   support

        None       0.90      0.96      0.93       964
     Failure       0.78      0.59      0.67       241

    accuracy                           0.88      1205
   macro avg       0.84      0.78      0.80      1205
weighted avg       0.88      0.88      0.88      1205

Task-level report:
ACC    0.885
AUC    0.897
F1     0.673
dtype: float64
```

Figure 1 shows the precision-recall curve. The average precision is 0.761.

The orange colored curve in Figure 2 shows the ROC curve. The AUC is 0.90.

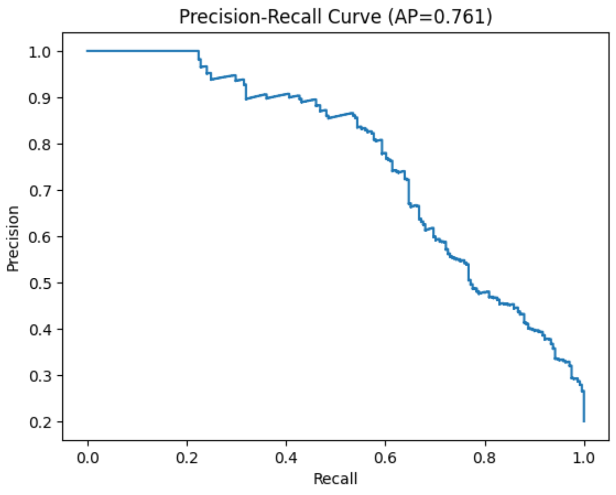


Figure 1: Precision Recall Curve

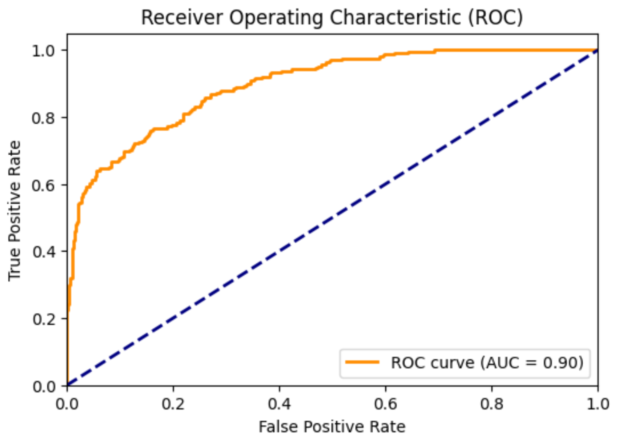


Figure 2: ROC and AUC

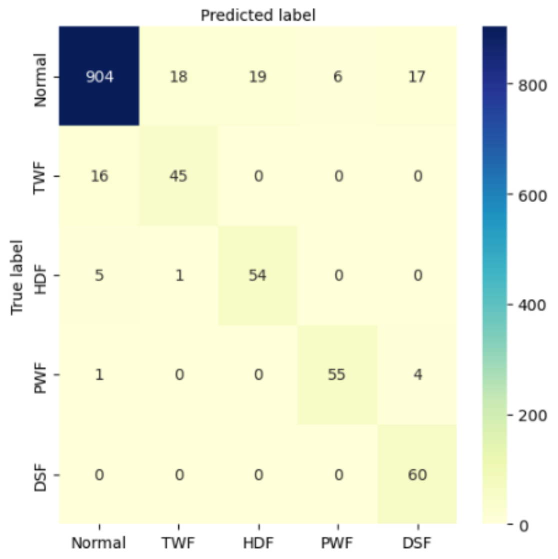
3.2 An Example of Multiclass Classification

Let's take a look at the multiclass classification problem in the Python Notebook. The following is the result using a QDA classifier.

```

 Class-level report:
              precision    recall  f1-score   support

      Normal       0.98      0.94      0.96       964
         TWF       0.70      0.74      0.72        61
         HDF       0.74      0.90      0.81        60
         PWF       0.90      0.92      0.91        60
         DSF       0.74      1.00      0.85        60

    accuracy                           0.93      1205
   macro avg       0.81      0.90      0.85      1205
weighted avg       0.94      0.93      0.93      1205

Task-level report:
ACC    0.928
AUC    0.983
F1     0.930
dtype: float64
```

The confusion matrix is in Figure 3 .

Figure 3: Confusion matrix for the testing result on MNIST dataset



References

[1] Yaser S Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning From Data*, volume 4. AMLBook New York, 2012.

[2] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[3] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[4] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[5] Larry Wasserman. *All of statistics: a concise course in statistical inference*, volume 26. Springer, 2004.